Machine, Data And Learning

Assignment 1 Report

Team

1. Vishal Reddy Mandadi 2019101119
2. Hrishi Narayanan 2019113022

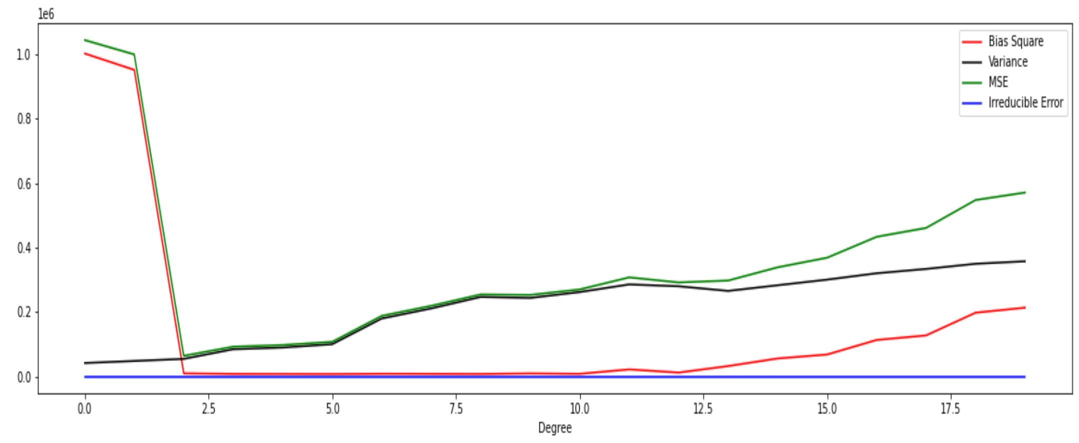
Task report

Note: The graph and other tabulated values were generated at the time of submission. These values could see minor deviation if retrained (during the evals), however the pattern/trend will be similar. We could have used some specific seed in order to make the code generated graphs reproducible but we haven't as we weren't asked to.

Task 1

The function LinearRegression.fit() is used to train the model on a dataset sent to this function as an argument. The fit method first conducts forward propagation i.e. it calculates the output of the model given the training set as input. It then compares the output with the actual values given in the dataset. Based on the difference/error it performs gradient descent (more like a hill climbing search and more details depend on the optimizer chosen) in order to reduce the difference between the output and the actual values in the next iteration. This process is continued until we reach a point (global minimum) where the difference will no longer decrease with parameter changes. Essentially, the function tries to find the best set of parameters that makes the model the closest approximation of the actual mapping (from input to correct output) through hill climbing search in general and gradient descent algorithm (as a default optimizer) in particular with error or loss or objective function as mean squared error function unless it is specifically provided to use a different one.

Task 2



Model Complexity (Polynomial degree)	Bias	Variance
1	230.999083	41663.230654
2	223.757672	47962.386589
3	-15.809085	54421.427071
4	-9.393410	84466.677734
5	-8.069212	89719.244173
6	-6.653591	99970.097742
7	-3.956608	179752.804217
8	-6.541790	211112.288410
9	-4.912845	246798.383011
10	-6.995527	243656.535835
11	-9.479391	261897.356331
12	0.796650	285622.804385
13	-14.601586	279825.165794
14	-20.721083	265450.656354
15	-30.135992	282991.343896
16	-30.277097	300446.187575

17	-43.200015	320206.859537
18	-45.594528	333621.861213
19	-59.872162	349686.160680
20	-63.775803	357339.413903

The complexity of the model increases with increase in the highest degree of the polynomial features. From the table it is clear that the bias of the model decreases with increase in the complexity of model while variance of the model increases with the increase in the complexity. We can further see that the sensitivity of bias to the change in complexity is exponential in nature upto degree 3. The bias is also found to be slightly higher for last few degrees than the intermediate degrees in the graph.The following points will explain the behaviour:

1. Variance vs complexity:
    a. Higher degree polynomials implies higher number of features. When we have a training dataset, higher degree polynomials would overfit the training data i.e. they also learn and model noise in the given training data set. Due to this, the same degree polynomial shows varying performance on test set the predictions depend highly on the data set it is trained on and the noise it contains (as the different datasets have different noise, therefore the models (trained on different data sets) perform differently on test sets as they have modeled themselves to include noise which is different from each other). Therefore, we see huge variance in the test set predictions given by the same higher polynomial but trained on different datasets. The lower degree polynomials, on the other hand, won't even be able to model the target function properly, so modelling noise would be much more difficult. Since they pick up very few features from the dataset, we see that they neither perform well on train sets nor the test set. And their performance on test set won't change much on changing the data set as it captures the much general features while leaving out the specific ones (which are caught by higher degree polynomials)
2. Bias vs complexity:
    a. Higher degree polynomials will be able to generate or model more complex mappings from input to output than lower degree polynomials. This means that when the target function mapping input to output is complex, the higher degree polynomials can estimate the function with lower error than the lower degree polynomials. Therefore, we should see lower bias error (which is the difference between the outputs of target function and our estimator) in case of higher degree polynomials and high bias error in lower degree polynomials .
    b. Higher degree polynomials can also emulate the behaviour of any lower degree polynomial by substituting the coefficients of their higher degree terms as zeros. So, if the target function is not so complex, a lower degree polynomial would be sufficient to model it so the lower degree polynomial will give lower bias error. Since, the higher degree polynomial can emulate lower degree polynomials as

well, the bias error will be low even in this case. Additionally the higher degree polynomials might even model the noise that comes with the input data and this would lead to much lower bias error. That is the reason why higher degree polynomials are still able to show lower error although the difference is negligible

Task 3

Degree of polynomial model	Irreducible error
1	-1.164153e-10
2	-2.328306e-10
3	0.000000e+00
4	0.000000e+00
5	0.000000e+00
6	1.455192e-11
7	0.000000e+00
8	-8.731149e-11
9	-5.820766e-11
10	-2.910383e-11
11	-5.820766e-11
12	-5.820766e-11
13	-5.820766e-11
14	-5.820766e-11
15	0.000000e+00
16	-5.820766e-11
17	1.164153e-10
18	0.000000e+00
19	2.328306e-10
20	-1.164153e-10

The average irreducible error calculated for each model was of order 10^{-10}. Irreducible error does not depend on the model, it depends on the noise in test data. Since the test data is common for all polynomial models, we expect irreducible error to be a constant for all models. But some variance is clearly seen in the graph. The variance though, that we observe in this case, is negligible and this could be a floating point problem. So generally, the computers calculate the solutions upto a certain precision and round the remaining off. This would lead to noise and this noise is the one that appears as variance in the graph.

Task 4

Underfitting and overfitting regions can be identified by comparing the total error plot with each of them. The global minimum of “Total error” plot corresponds to the ideal fit model. All those to the left of the global minimum are said to be underfit and all those to the right are said to be overfit (but with varying levels of underfitting or overfitting respectively). From the graph, the model with degree 3 gives a minimum total error. The model with degree 4 gives similarly small total error. Therefore, we can say that polynomials of degree 3 and 4 are fit ideally to the train data. While the polynomial models with degrees 1 and 2 are underfit and the polynomial models with degrees greater 4 are overfit to the train data.

The given data seems to have a relatively simple mapping from input to output points. Therefore a degree 4 polynomial was more than sufficient to model the target function that describes the data.